Remedy Year 2000 Compliance Verification

Testing Methodology

This document describes the Year 2000 (Y2K) compliance testing methodology for the Remedy® Action Request System®. This document is provided for informational purposes only; it is not intended to expand or modify any terms of Remedy's standard software license agreement or specific customer agreement(s). For a copy of the most current product compatibility matrix and periodic updates to Remedy's Year 2000 compliance documents, visit Remedy's web site at http://www.remedy.com.

Remedy software currently supports any date value set by the operating system, with the exception of Julian and 'year with day offset within the year' dates. All date information, such as creation dates, modification dates, and time/date stamps, is generated using the date value provided by the operating system (Microsoft® Windows®, Apple® Macintosh®, UNIX®, and others).

Test Methodology Description (Revised: April 2, 1999)

Date Formats
Date format testing explicitly includes the valid symbols as specific tests—such as slash (/) and hyphen (-) for U.S., slash (/) for Japan, and period (.) for Europe. Localized versions of AR System have been tested for month abbreviations.

Valid Operating System Values
The correct date range has been tested for each operating system by platform and by version. For example, the AR System running on Solaris has been tested for dates from 1/1/70 to 1/1/2038.

Certain operating systems may not produce correct values, and Remedy products depend on operating systems that are year 2000 compliant.

Valid Ranges for Clients
All Remedy client tools have been tested for their ability to correctly handle the valid date ranges, as follows:

- Any date value before 1/1/70 should generate an error and not allow the value to be entered.
- Any date value after 12/31/2036 (NT) or 1/1/2038 (UNIX) should generate an error and not allow the value to be entered.

Break Workflow
Active link and filter functionality has been tested to ensure that it can correctly handle dates after the year 2000.

Tests executed include users and administrators doing queries that attempt to circumvent normal date range checking. In addition, all 16-bit clients have been tested for broken workflow.

Specific Tests for Boundary Crossing
The product has been carefully tested to ensure correct handling of the transition from 1999 to 2000. This boundary testing focussed on what will happen at midnight 12/31/99.

The product has also been tested to ensure that the year 2000 is handled as a leap year, and that the year 2001 is not handled as a leap year. This boundary testing investigated what will happen at midnight 12/31/99, Feb 29, 2000, and Mar 1, 2000.

Testing Year 2000 with the Action Request System

Test Type	Test Description	Results
Create Schema with Default Times using the year 2000	Create a schema with a date/time field, and save changes Set the default value of the date/time field to 10/1/00. Without closing the field property dialog, save the schema; and verify that the date/time format is preserved. Repeat, using long date formats	Pass
Using Year 2000 in an active link, filter, and escalation	Create an active link, filter, and escalation with an action to set a date/time field with a date greater than or equal to the year 2000 Create a filter that will set a date/time field with the keyword $TIMESTAMP$, and add four years Create a filter that will send a notification; use keywords with time values Create an escalation that will set a date/time field with the following value: $Date/Time$ - $CreateDate$/60, and verify result Create an escalation that starts at 12/31/99 11:45pm, and execute every five minutes. Verify correct time stamp	Pass
Date Formats to Submit and Query: mm/dd/yy (12/01/00) mm/dd/ccyy (12/01/2000) dd/mm/yy(01/12/00) dd/mm/ccyy(01/12/2000) dd/mm (day and month only) Complete date (December 31, 1999) Date, abbreviated month (Dec 1, 1999)	Open the User Tool, and perform a submit and query using the different formats for date/time fields: Verify that all date/time fields display the date/time correctly in display modify windows (for example, Modified Date, Create/Date, and so on)	Pass
Date Sorting	In the User Tool, change the sorting order for date/time fields. Run a query, open the query list, and report and verify that the sorting is correct Change the sorting in a custom AR System Report, and verify that these sort options override the default sorting	Pass
Default Dates	Admin Defined Dates are displayed properly when admin defaults are set in User Tool Set User Tool default dates. Verify that dates are set correctly.	Pass

Leap Year Computations	Submit, Query, and Modify using dates: 1996, 2000, 2004 (and so on), using 2/28 and 2/29	Pass
Special Dates	Submit, Query, and Modify using dates: December 31, 1999 January 1, 1999 March 1, 1999 April 9, 1999 July 1, 1999 August 22, 1999 September 9, 1999 October 1, 1999 January 0, 2000 January 1, 2000 January 3, 2000 January 10, 2000 February 28, 2000 February 29, 2000 March 1, 2000 October 10, 2000 January 1, 2001 February 29, 2001 February 29, 2004 January 1, 2010	Pass
Leading zeros in the day	Submit, Query, and Modify using days: 01, 02, 09 (12/01/99 versus 12/1/99)	Pass
Leading zeros in the month	Submit, Query, and Modify using months: 01, 02, 09 (01/15/36 versus 1/15/36)	Pass
Can day numbers exceed actual?	Submit, Query, and Modify using dates: May 32, February 30 (and so on)	Pass
Date Comparisons	Use Queries that add or subtract time to give results of earliest and latest permissible dates. Example: **'Create-date' >= $TIMESTAMP$ - <number of seconds>** **Calendar is the same every 28 years** Submit, using current date, and then change the date 28 years after (day and date should be identical)	Pass
Dates Set by Active Links, Filters, and Escalations	Set Fields (Date, keywords, Date from different fields and schemas) calculations that set results in integer fields	Pass
Dates displayed in notifications	Time stamp of notification Date/time included in the message	Pass
Macros	Include date/time fields in parameters using permissible dates; for example, Date/Time, Create Date, and Modified Date (and so on)	Pass
Reporting	Dates are displayed correctly. Send a report to a file, printer, and screen	Pass

Statistics	Modify status on multiple entries to create a report using statistics with the Status History Fields	Pass
	Use other combinations of date/time fields to calculate (sum, average, minimum, maximum)	
Importing Data	Using the Import Tool, import each record into the same schema by using the date time formats	Pass
Test Environment	Change the date on the server to 2001, and repeat above tests	Pass